

تمرین ۲ درس طراحی الگوریتم

مریم سادات هاشمی
سید صالح اعتمادی

دانشگاه علم و صنعت ۹۸-۹۷

لطفاً به نکات زیر توجه کنید:

- مهلت ارسال این تمرین شنبه یازدهم اسفند ساعت ۱۱:۵۹ ب.ظ است.
- این تمرین شامل سوال های برنامه نویسی می باشد، بنابراین توجه کنید که حتماً موارد خواسته شده را رعایت کنید.
- نام شاخه، پوشه و پول ریکوست همگی دقیقاً A2 باشد.
- در صورتی که به اطلاعات بیشتری نیاز دارید می توانید با ایدی تلگرام @maryam_sadat_hashemi در ارتباط باشید.
- اگر در حل تمرین شماره ی ۲ مشکلی داشتید، لطفاً به این [لینک](#) مراجعه کنید و زمانی را برای رفع اشکال تنظیم کنید.

موفق باشید.

توضیحات کلی تمرین

تمرین این هفته ی شما، ۲ سوال دارد که باید به همه ی این سوال ها پاسخ دهید. برای حل این سری از تمرین ها مراحل زیر را انجام دهید:

۱. ابتدا یک پروژه به نام A2 بسازید.
۲. کلاس هر سوال را به پروژه ی خود اضافه کنید و در قسمت مربوطه کد خود را بنویسید. هر کلاس شامل دو متد اصلی است:
متد اول: تابع solve است که شما باید الگوریتم خود را برای حل سوال در این متد پیاده سازی کنید.

متد دوم: تابع process است که در TestCommon پیاده سازی شده است. بنابراین با خیال راحت سوال را حل کنید و نگران تابع process نباشید! زیرا تمامی پیاده سازی ها برای شما انجام شده است و نیازی نیست که شما کدی برای آن بنویسید.

۳. اگر برای حل سوالی نیاز به تابع های کمکی دارید؛ می توانید در کلاس مربوط به همان سوال تابع تان را اضافه کنید.

اکنون که پیاده سازی شما به پایان رسیده است، نوبت به تست برنامه می رسد. مراحل زیر را انجام دهید.

۱. یک UnitTest برای پروژه ی خود بسازید.
۲. فولدر TestData که در ضمیمه همین فایل قرار دارد را به پروژه ی تست خود اضافه کنید.

۳. فایل GradedTests.cs را به پروژه ی تستی که ساخته اید اضافه کنید. توجه کنید که لازم نیست که برای هر سوال TestMethod بنویسید. تمامی آنچه که برای تست هر سوالتان نیاز دارید از قبل در این فایل برای شما پیاده سازی شده است.

در این تمرین، علاوه بر فایل هایی با پسوند txt که تست کیس های سوالات هستند و شما از آن ها برای تست کدتان استفاده می کنید، فایل هایی با پسوند webgraphviz نیز در پوشه TestData وجود دارد. شما با استفاده از این فایل ها می توانید گراف های کوچکتر از ۱۰۰ گره را به صورت تصویری در این [سایت](#) مشاهده کنید.

۱ محاسبه حداقل تعداد قسمت های پرواز

فرض کنید که شما می خواهید حداقل تعداد بخش های پرواز لازم برای رفتن از یک شهر به شهر دیگر را محاسبه کنید. برای این منظور یک گراف غیر جهت دار در نظر بگیرید که: رأس ها شهرها را نشان می دهند، و اگر پروازی بین دو شهر وجود داشته باشد، یک یال بین رأس های متناظر با این شهر ها قرار داده می شود. سپس، کافی است که کوتاه ترین مسیر از شهر مورد نظر به سوی شهر دیگر را در این گراف غیر جهت دار پیدا کنید.

بنابراین شما باید الگوریتمی بنویسید که کوتاه ترین مسیر موجود بین رأس u و رأس v را در گراف غیر جهت دار با n رأس و m یال را محاسبه کند. (یعنی حداقل تعداد لبه ها در مسیر از u به v).

خط اول فایل ورودی، تعداد رأس های گراف را نشان می دهد و هر یک از خطوط بعدی شامل دو عدد می باشد که بیانگر وجود یک یال بین این دو عدد(رأس) است. در خط آخر هم دو عدد وجود دارد که باید کوتاه ترین مسیر بین عدد(رأس) اول و عدد(رأس) دوم را بدست آورید. اگر کوتاه ترین مسیر بین دو رأس مورد نظر وجود داشته باشد، در فایل خروجی طول مسیر قرار می گیرد و در غیر این صورت عدد -1 درج می شود.

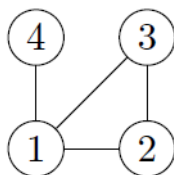
نمونه ۱

ورودی:

4
1 2
4 1
2 3
3 1
2 4

خروجی:

2



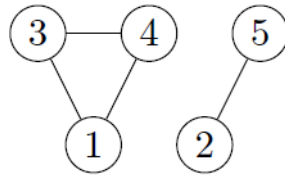
نمونه ۲

ورودی:

5
5 2
1 3
3 4
1 4
3 5

خروجی:

-1



۲ بررسی دو طرفه بودن یک گراف

یک گراف غیر جهت دار دو طرفه نامیده می شود اگر بتوان رأس های آن را به دو بخش تقسیم کرد به طوری که هر لبه گراف به رأس ها از قسمت های مختلف متصل شود. به زبان ساده تر، یک گراف دو طرفه است، اگر رأس های آن را بتوان با دو رنگ (به عنوان مثال سیاه و سفید) رنگی کرد به طوری که نقاط انتهایی هر لبه رنگ های متفاوتی داشته باشند. گراف های دو طرفه به طور طبیعی در برنامه های کاربردی بوجود می آیند که در آن گراف برای مدل سازی ارتباط بین اشیاء دو نوع مختلف (مثلا پسر و دختر، یا دانش آموز و خوابگاه) استفاده می شود.

در این سوال، یک گراف غیر جهت دار با n رأس و m یال داده می شود و شما باید برنامه ای بنویسید که مشخص کند آیا این گراف دو طرفه است یا خیر. خط اول فایل ورودی شامل یک عدد است که نشان دهنده ی تعداد رأس های گراف غیر جهت دار است و در هر یک از خطوط بعدی دو عدد وجود دارد که بیانگر وجود یک یال یا لبه بین عدد(رأس) اول و عدد(رأس) دوم است. اگر گراف داده شده دو طرفه باشد، خروجی یک و در غیر این صورت صفر خواهد بود.

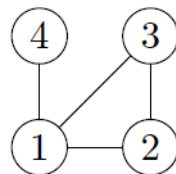
نمونه ۱

ورودی:

4
1 2
4 1
2 3
3 1

خروجی:

0



نمونه ۲
ورودی:

5
5 2
4 2
3 4
1 4

خروجی:

1

